

● Een introductie in Basic ●

Gordon Rahman

Inleiding

Even een kort speciaal voorwoord van mij, René Suiker, als introductie op onderstaande tekst. We zijn bij CompUsers vooral van de toepassingen, de applicaties, maar met zaken als WebOntwerp, Scratch, GameOntwerp, begint de grens tussen programmeren en toepassingen langzaam te vervagen.

Daarnaast zijn we als CompUsers altijd gastvrij geweest naar andere groeperingen en nu de SoftwareBus zachtjes aan een officieel HCC Orgaan aan het worden is, leek het ons (bestuur, redactie) wel een goed idee om ook bij andere interessegroepen kopij te verwerven.

En door de goede contacten die we al hebben met de IG Programmeren kwam het zo aan de orde, dat Gordon Rahman van deze groepering zelfs graag een stukje over BASIC wil schrijven in onze SoftwareBus.

In dat kader dus deze bijdrage aan de SoftwareBus. Hierbij het woord aan Gordon.

Wie is Gordon?

Ik ben dus Gordon Rahman. Ik ben al jaren enthousiast hobbyprogrammeur en speel een belangrijke rol bij de BASIC-activiteiten van de IG Programmeren. In dat kader ben ik blij met de gelegenheid om mijzelf, mijn IG en vooral mijn hobby in de SoftwareBus te kunnen presenteren. Ik was vroeger een 'vraagbaak' voor TRS80 Basic voor de computers van Tandy / Radio Shack en later voor Liberty BASIC voor Windows. Sinds 2016 verzorg ik jaarlijks een Workshop Liberty BASIC voor de Regio Amsterdam en sinds de oprichting van de Programmeren IG doe ik dat ook voor de Programmeren IG. Op elke CompUfair zijn wij aanwezig als BASIC deelgroep van de IG Programmeren.

BASIC

Er wordt vaak wel denigrerend gedaan over BASIC, het zou een beginnerstaal zijn, je kunt er niet gestructureerd mee programmeren, het is een heel beperkte taal, maar dat ga ik in dit artikel allemaal weerleggen. Ik ga uitleggen dat BASIC een volwassen programmeertaal is, waar je in principe alles mee kunt bouwen. Net als ELKE andere taal heeft BASIC zijn sterke en zwakke kanten en daar ga ik niet geheimzinnig over doen. Ik ga wel uitleggen waarom ik enthousiast ben over BASIC. Maar alvorens helemaal de BASIC in te duiken, eerst even een stapje terug.

Programmeren

Op Wikipedia vind je een uitgebreide uitleg over wat programmeren feitelijk is. En dan heb ik het over programmeren van computers, want je hebt ook andere programma's, zoals het programmeren van NPO 1 op dinsdagavond, maar dat laten we nu even buiten beschouwing.

De definitie volgens Wikipedia: *'Programmeren is het schrijven van een computerprogramma, een concrete reeks instructies die een computer kan uitvoeren.'* Dit is de taak van een softwareontwikkelaar of programmeur. Programmeren wordt in het algemeen niet direct in machinetaal gedaan, maar in een programmeertaal. De programmacode die wordt geschreven heet broncode en wordt door een assem-

bler, compiler of interpreter omgezet in machinecode. Met name in het verleden werd voor 'programmeren' ook 'coderen' gebruikt.

Kortweg komt het er op neer, dat je een reeks instructies schrijft om de computer te laten doen wat je wilt. Echter, een computer kan een aantal dingen heel goed en vooral heel snel, maar kan ook een aantal dingen niet. En zeker niet uit zichzelf. Want met een programma doet de computer niet per definitie wat je wilt, maar wat je hem opdraagt. En dat legt de bal bij jou, je moet heel nauwkeurig zijn in wat je de computer opdraagt, anders gebeuren er onverwachte dingen. In de artikelen over Scratch zijn daarover al wat voorbeelden voorbij gekomen. Maar als het je dan wel lukt om de computer te laten doen wat je wilt, geeft dat een goed gevoel.

Programmeertalen

Er zijn enorm veel programmeertalen en er zijn vele manieren om ze onder te verdelen. Een bekende manier is de onderverdeling in generaties. De eerste generatie is dan machinetaal, toen er nog niets anders was werd de computer aan het console met schakelaars geprogrammeerd. Dat was een heel karwei en je had het niet zo snel door als je ergens een foutje maakte. De tweede generatie is dan assembler, dat is in feite stap voor stap machinecode, maar dan leesbaar neergezet. De derde generatie zijn dan de talen, die al minder afhankelijk waren van de machine waarop ze draaiden. In min of meer leesbare vorm werden instructies uitgeschreven, die later vertaald werden in machinecode om uitgevoerd te worden. Hiertoe rekenen we procedurele talen als COBOL, C, Pascal en dus ook BASIC. Daarnaast worden de objectgeoriënteerde talen zoals Java en C++ ook als 3^e-generatietaal gezien.

De 4^e-generatietalen zijn talen met een hoger abstractieniveau en voor speciale doelen ontwikkeld. Voor business-databases wordt er bijvoorbeeld veel gebruik gemaakt van SQL. Er zijn intussen ook 5^e-generatietalen, maar daar blijven we in dit kader maar even bij weg. We richten ons dus nu op BASIC, een 3GL- taal (3^e-generatie dus)-. Maar ook het nu zo populaire Python is een 3GL-taal, dus de leeftijd is niet bepalend.

Zoals gezegd, over de theoretische basis van programmeren is op Wikipedia heel veel achtergrondinformatie te vinden, en uiteraard ook in vele bibliotheken; het is nuttig om even een kader te schetsen, maar ik wil er niet te lang bij stilstaan.

Even wat terminologie

Zoals ook al gezegd voeren computers een programma uit. Een programma, bijvoorbeeld geschreven in BASIC, noem ik ook wel een listing. Soms noem ik het ook wel een script.

Waarom BASIC

BASIC is een acroniem voor **B**eginners **A**ll-purpose **S**ymbolic **I**nstruction **C**ode. Het is een taal die bedoeld is voor algemeen gebruik, dus niet voor een specifieke toepassing, het is een programmeertaal, dus geen natuurlijke taal, het heeft een welomschreven syntax en het is een taal waarmee je makkelijk kunt leren programmeren.

BASIC lijkt voor mij de eenvoudigste weg te zijn om te leren programmeren. In BASIC leer je de concepten van het programmeren onder de knie te krijgen en als je eenmaal kunt programmeren, dan kan je in principe elke andere taal ook leren.

Zelf maak ik wel onderscheid tussen imperatieve talen en objectgeoriënteerde talen. De aanpak bij een objectgeoriënteerde taal is wezenlijk anders, omdat je daar de objecten en de bewerkingen strikt gescheiden houdt, terwijl je in imperatieve talen de verwerking en de gegevens door elkaar heen laat lopen.

Bij BASIC schrijf je de code voor zowel de objecten (ook daar kan ik uren over praten, maar zie het maar even als de data van het programma) als de bewerkingen, dus in één programma (listing).

Het voordeel van deze aanpak is dat je snel ziet wat een programma doet, de code is eenvoudig leesbaar. Het programmeren erin geeft mij plezier: je moet alles zelf schrijven en je bouwt kennis op die nooit verloren gaat.

Je ziet dan ook vaak dat mensen die in BASIC programmeren alles zelf doen (of met een team van mensen), terwijl je in andere talen vaak deeloplossingen kunt kopen. Dat kan in BASIC ook, en in andere talen kun je ook alles zelf schrijven, maar dat is niet de gebruikelijke manier.

Als je direct met Windows communiceert doe je dat via DLL-bestanden. Dit zijn in feite deeloplossingen die door anderen bedacht zijn, maar om direct met Windows te communiceren kan je hier niet omheen.

De eenvoudigste 'BASIC voor Windows' die ik ken is Liberty BASIC (voor Windows) en daar ga ik me verder toe beperken. Het maakt gebruik van de DLL's van Windows en daarom kan het ook heel direct met Windows werken. Dat is iets dat in oude versies van BASIC nog niet kon: die waren in feite op regel-output gebaseerd.

Liberty BASIC stamt uit de tijd dat de com-poorten nog door BASIC werden bestuurd. Daarom kan Liberty BASIC dat nog altijd, terwijl Windows uw hardware afschermt met speciale drivers. Dus je kunt zowel via de DLL's werken, als de poorten rechtstreeks benaderen. Dit kan nuttig zijn als je bijvoorbeeld Arduino's e.d. wilt besturen.

Deze BASIC lijkt op de BASIC van de Commodore-Amiga, aangevuld met 42 extra commando's om de Windows-vensters en grafische mogelijkheden te benutten. Zo ziet het resultaat van een listing om een venster met daarin een knop te tonen er bijvoorbeeld uit:

```
Liberty BASIC v4.5.1 (Gordon Rahman) - C:\Users\Gordon\Desktop\softwarebus1.bas
File Edit Run Setup Help
WindowWidth = 800
WindowHeight = 600

button #wdown.knop1 "Eerste knop", [knopNummer1], UL, 50, 50
Open "Mijn eerste venster in Liberty BASIC" for window as #wdown
wait

[knopNummer1]
print "Deze tekst is in de mainwin (console mode) te zien"
wait

rem voor niet uit te voeren opmerkingen (remark)
' een apostrof geldt ook als rem
'ALF+F4 geeft natuurlijk in Windows
'het automatisch afsluiten van het actuele venster

[quit]
close #wdown
end
```

Liberty BASIC kent net als vele andere talen de volgende constructies:

1. Lus-structuren
(FOR TO STEP-NEXT)
2. Keuze-structuren
(IF THEN)
(IF THEN-ELSE-END IF)
(SELECT CASE-CASE ELSE-END SELECT)
3. Herhaal-structuren
(WHILE-WEND)
(DO)
(DO UNTIL)
(DO WHILE LOOP)
(DO LOOP UNTIL)
(DO LOOP WHILE)
4. Boomstructuren
(GOTO)
(GOSUB-RETURN)
(FUNCTION-END FUNCTION)
(SUB-END SUB)

In elke computertaal worden andere commando's en speciale karakters gebruikt. De syntax verschilt, maar de overstap van de ene taal naar de andere is qua syntax geen grote. Computertalen verschillen voornamelijk op de manier waarop zij hun variabele gegevens bewaren. Daarmee bedoel ik dat sommige talen (vooral de moderne) van de programmeur eisen dat die steeds en overal aangeeft wat voor soort en hoe variabelen bewaard moeten worden. De talen vereisen bijvoorbeeld in elk object, script, struct, e.d. dat alle variabelen vooraf gedeclareerd worden. Liberty BASIC en ook de nieuwste Visual-BASIC houden dat zelfstandig bij. Liberty BASIC houdt ook zelfstandig in de gaten welke nauwkeurigheid voor getallen aangehouden moet worden. Soms klaag ik erover dat Liberty BASIC de waarde van Pi (3.14xxxxxxxxxxxx) slechts tot 12 cijfers achter de komma nauwkeurig hanteert. Maar als je een financiële toepassing bouwt, dan hoef je eigenlijk maar 3 cijfers achter de komma te gebruiken.

Ik hoop dat ik hiermee wat nieuwsgierigheid heb gewekt bij mensen die in het verre verleden stopten met zelf programmeren (omdat alles toch al bestond, enz.). De BASIC enthousiastelingen van de Workshop in Amstelveen hebben in de afgelopen jaren tientallen programma's gemaakt. Vele toepassingen (in Liberty BASIC geschreven) zijn dagelijks gebruikte 'juwelen' zoals bijv. die in de melkveehouderijen in Nederland, bij orgelbouwers en bij wiskunde op sommige middelbare scholen.

Maar het meeste plezier hebben we van onze eigen listings. De Workshop-deelnemers hebben vele tientallen listings geschreven die ze graag met plezier willen demonstreren.

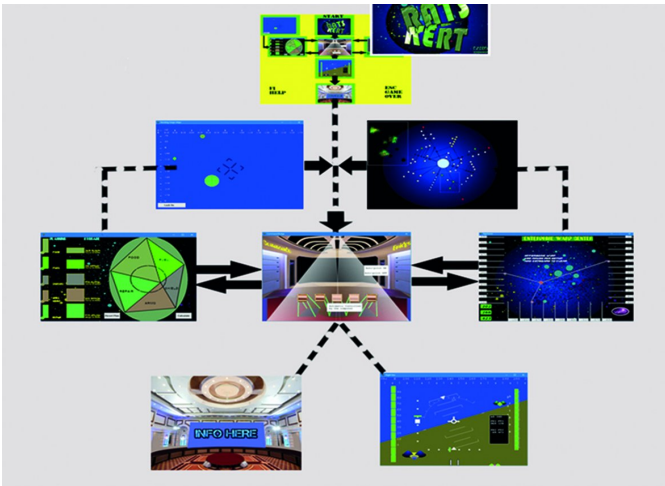
De Liberty BASIC workshop.

De HCC regio Amsterdam biedt de Liberty BASIC groep reeds jaren ruimte voor een Workshop Liberty BASIC. Dit artikel is een kennismaking met de Workshop, Liberty BASIC en een kennismaking met de Programmeren IG Basic Groep.

Op de volgende pagina staat een plaatje van het Helpschermb van de game die tijdens de Liberty BASIC Workshop 2020-2021 is gemaakt.

De Programmeren IG geeft jaarlijks een Workshop Liberty BASIC in samenwerking met de Afdeling Amsterdam (in wijkcentrum de Meent te Amstelveen).

In verband met de Covid-19-perikelen is de workshop van 2020 door middel van Online sessies verder gegaan. De Programmeren IG heeft daarnaast wekelijks vier online sessies voor Liberty BASIC, Visual BASIC, Python en Games Programmeren met UNITY. Dat laatste sluit aan bij wat CompUsers nu



Helpscherm

ook probeert op te zetten, alleen zijn we bij IG Programmeren intussen zo veel verder, dat de aansluiting nog niet te maken valt. Hopelijk gaat dat met BASIC beter lukken.

Een eigen game

Tot slot nog een momentopname uit de game die in 2020-2021 door de LB Workshop samen is geproduceerd. Maandelijks vindt er een 'plenaire' sessie (waarbij alle deelgroepen van de IG aanzitten) plaats in De Bilt. Ook deze bij-



Fight screen

eenkomsten zijn i.v.m. Covid-19 omgezet tot online sessies. Bij de IG Programmeren zijn er verschillende 'BASIC-groepen': Visual-BASIC, PowerBASIC, enz. Daarnaast kun je er ook terecht voor computertalen als Python, Forth, Pascal, LISP, MSX-Basic, enz.

Alle BASIC programmeurs (amateurprogrammeurs) streven ernaar om wat van hun vaardigheden te demonstreren met BASIC voor Windows en of andere besturingssystemen. CompUsers heeft nog geen BASIC-groep, maar daar wil ik aan werken.